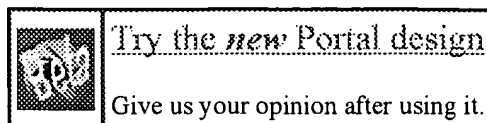



[> home](#) [> about](#) [> feedback](#) [> login](#)

US Patent &amp; Trademark Office



## Search Results

Search Results for: **["simd within a register"]**  
 Found **9** of **127,944** searched.

## Search within Results


[> Advanced Search](#) [> Search Help/Tips](#)


---

**Sort by:** Title Publication Publication Date Score Binder
 

---

**Results 1 - 9 of 9** short listing
 

---

**1** A C compiler for a processor with a reconfigurable functional unit 85%


Zhi Alex Ye , Nagaraj Shenoy , Prithviraj Baneijee

**Proceedings of the 2000 ACM/SIGDA eighth international symposium on Field programmable gate arrays** February 2000

This paper describes a C compiler for a mixed Processor/FPGA architecture where the FPGA is a Reconfigurable Functional Unit (RFU). It presents three compilation techniques that can extract computations from applications to put into the RFU. The results show that large instruction sequences can be created and extracted by these techniques. An average speedup of 2.6 is achieved over a set of benchmarks.

**2** High-cost CFD on a low-cost cluster 84%


Thomas Hauser , Timothy I. Mattox , Raymond P. LeBeau , Henry G. Dietz , P. George Huang

**Proceedings of the 2000 ACM/IEEE conference on Supercomputing (CDROM)** November 2000

Direct numerical simulation of the Navier-Stokes equations (DNS) is an important technique for the future of computational fluid dynamics (CFD) in engineering applications. However, DNS requires massive computing resources. This paper presents a new approach for implementing high-cost DNS CFD using low-cost cluster hardware. After describing the DNS CFD code DNSTool, the paper focuses on the techniques and tools that we have developed to customize the performance of a cluster ...

**3** Reception and posters: Programming portable optimized multimedia 80%


applications

Juan Carlos Rojas , Miriam Leeser

**Proceedings of the eleventh ACM international conference on Multimedia** November 2003

Multimedia computer architectures can speed-up applications significantly when programmed manually. Optimized programs have been non-portable up to now, because of differences in instruction sets, register lengths, alignment requirements and programming styles. We solve all these problems by using a library of C pre-processor macros called MMM. We implemented three examples from video compression in MMM, and automatically translated them into optimized code for four distinct multimedia processor ...

**4** Embedded tutorial: Code generation for embedded processors 80%


Rainer Leupers